HOJA Nº 1

FECHA

2) QUIERO UN CAMINO DE s a t QUE PASE POR r MÍNIMO ENTRE LOS QUE TIENEN LONGITUD PAR.

DISEÑO EL DIGRAFO H, CUYOS VÉRTICES VAN A SER LOS MISMOS VÉRTICES DE G DUPLICADOS. VOY A LLAMAR A LOS DOS CONJUNTOS DE VERTICES V_P y V_I

V_P SERÁ EL CONJUNTO DE VÉRTICES QUE REPRESENTARÁ QUE, CUANDO RECORRIMOS DESDE s LLEGAMOS A ESE VÉRTICE CON UNA CANTIDAD PAR DE ARISTAS.

V_I ANÁLOGO PERO CON UNA CANTIDAD IMPAR.

FORMALMENTE:

$V_P = \{(v,p) \mid \forall v \in V(G)\}$ p ETIQUETA

$V_I = \{(v,i) \mid \forall v \in V(G)\}$ i ETIQUETA

AMBOS CONJUNTOS FORMARÁN UN GRAFO BIPARTITO H DONDE EXISTIRA UNA ARISTA ENTRE UN VERTICE DE CADA CONJUNTO SI LA ARISTA ORIGINAL EN G CONECTA DOS VERTICES CON DISTINTA PARIDAD

$$E = \{(v,x) \to (w,y) \mid v \to w \in E(G) \quad x \neq y\}$$

ENTONCES MI DIGRAFO H SERÁ

$$H = (V_P, V_I, E), \; r_1, r_2$$

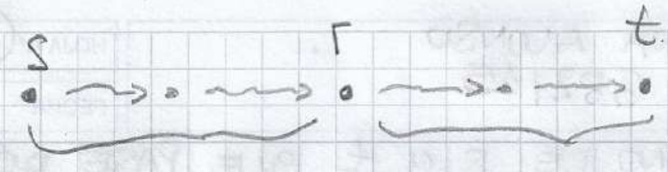
DONDE TENDRÉ DOS VÉRTICES IDENTIFICADOS r_1 y r_2 COMO DICE EL ENUNCIADO.

AHORA COMO QUIERO EL CAMINO DE s a t QUE PASE POR r DE LONGITUD PAR, PUEDO OBSERVAR QUE PUEDO DESCOMPONER EL CAMINO DE s a t COMO: EL MÍNIMO CAMINO DE s a r y MÍNIMO CAMINO DE r a t. ESTO LO SE POR LA PROPIEDAD VISTA QUE TODOS LOS CAMINOS INTERIORES SON MÍNIMOS.

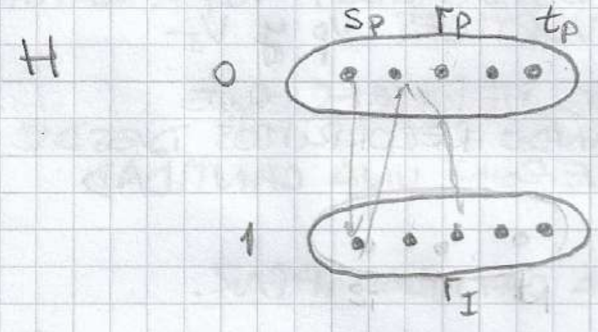
PUEDO OBSERVAR TAMBIEN QUE SI QUIERO QUE ESE CAMINO SEA DE LONGITUD PAR, TENGO DOS POSIBILIDADES:

- LOS SUBCAMINOS DE s a r y DE r a t TIENEN AMBOS LONGITUD PAR
- LOS SUBCAMINOS DE s a r y DE r a t TIENEN AMBOS LONGITUD IMPAR.

NOTA



par + par = par
impar + impar = par



SI LLEGO DESDE s a t EN LA PARTICIÓN PAR, TENGO UN CAMINO DE LONGITUD PAR. NECESITO VER QUE EXISTA UN CAMINO QUE PASE POR r.

NECESITO HALLAR ENTONCES:

a). UN CAMINO MÍNIMO DE s_p A r_p Y UN CAMINO MÍNIMO DE r_p A t_p PARA TENER DOS SUBCAMINOS DE LONGITUD PAR MÍNIMOS QUE FORMEN UN CAMINO MÍNIMO.

b). UN CAMINO MÍNIMO DE s_p a r_I Y UN CAMINO MÍNIMO DE r_I A t_p PARA TENER DOS SUBCAMINOS DE LONGITUD IMPAR QUE FORMEN UN CAMINO MÍNIMO PAR

OBS: SE QUE BUSCAR CAMINOS MÍNIMOS EN H CORRESPONDEN A CAMINOS MÍNIMOS DE G POR EXISTENCIA DE ESAS ARISTAS.

IDEA DEL ALGORITMO:

1 CORRO BFS EN H DESDE s_p PARA OBTENER LAS DISTANCIAS DESDE S. (EN PARTICULAR

2 CORRO BFS EN H DESDE t_p (INVIRTIENDO LAS ARISTAS PREVIAMENTE)

3 COMO TENGO LA DISTANCIA DESDE s_p HACIA TODOS LOS OTROS VÉRTICES, EN PARTICULAR LA TENGO A r_I Y r_p, Y TAMBIÉN TENGO LA DISTANCIA MÍNIMA DESDE ELLOS A t_p.
ENTONCES ME QUEDO CON EL MEJOR (MÍNIMO) ENTRE LAS POSIBILIDADES QUE VIMOS ANTERIORMENTE

$$\min \{ d(s_p, r_I) + d(r_I, t_p), d(s_p, r_p) + d(r_p, t_p) \}$$

NOTA

HOJA N° 2

FECHA

OBSERVACIÓN: LOS Γ_1 y Γ_2 MENCIONADOS EN EL ENUNCIADO CORRESPONDEN A s_p y t_p.

VEAMOS LA COMPLEJIDAD DEL ALGORITMO:

SEAN n y m LA CANTIDAD DE VÉRTICES y CANTIDAD DE ARISTAS DE G RESPECTIVAMENTE

$$V(H) = 2\,V(G) = 2n$$

$$E(H) = 2\,E(G) = 2m$$

EN ① CORRER BFS TIENE COMPLEJIDAD $O(2n+2m) = O(n+m)$

EN ② INVERTIR LAS ARISTAS ES $O(2m) = O(m)$ y CORRER BFS NUEVAMENTE $O(n+m)$

EL PASO ③ LO PUEDO VERIFICAR EN TIEMPO QUE SE QUE ES MENOR A $O(n+m)$

POR LO TANTO LA COMPLEJIDAD DEL ALGORITMO SERA DE $O(n+m)$

NOTA

HOJA N° 3

FECHA

3) QUIERO ENCONTRAR UN CICLO DE LONGITUD MÁXIMA EN EL GRAFO G. PARA ESO, QUIERO IDENTIFICAR TODOS LOS CICLOS DE G Y QUEDARME CON EL DE LONGITUD MÁXIMA.

PASOS:

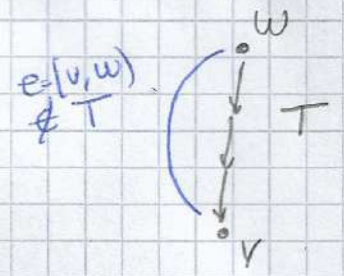
1) ENCONTRAR TODOS LOS CICLOS DEL GRAFO $\{c_1 \ldots c_n\} \in G$

2) VER SUS LONGITUDES Y QUEDARME CON EL CICLO QUE TENGA MÁXIMA LONGITUD.

$\Rightarrow$ G ES CACTUS Y TIENE A LO SUMO $O(n)$ ARISTAS

PARA EL PASO 1 CORRO DFS MANTENIENDO UN VECTOR E DONDE E_x INDICA SI LA ARISTA X YA PERTENECE A UN CICLO DE ARISTAS RECORRIDAS POR DFS, Y EL VECTOR INICIALMENTE INICIA TODO EN FALSO.

LUEGO DE EJECUTAR DFS OBTENGO UN ARBOL T, DONDE SI ANALIZAMOS $E(G)/E(T)$ TENEMOS LAS ARISTAS BACK-EDGE ~~TAL~~ QUE:



SI ENCONTRAMOS UNA BACK-EDGE, ENTONCES w ES UN ANCESTRO DE v EN T, POR LO TANTO EL CAMINO DE w A v CON LA ARISTA e FORMA UN CICLO EN G.

MARCAMOS TODAS LAS ARISTAS QUE PERTENEZCAN A UN MISMO CICLO ANALIZANDO LAS ARISTAS BACK-EDGE. PARTICIONAMOS LAS ARISTAS CON EL MISMO IDENTIFICADOR PARA SABER CUALES SON LOS CICLOS.

EN EL PASO 2 RECORREMOS CADA CICLO Y CONTAMOS LA CANTIDAD DE ARISTAS QUE PERTENECEN, MANTENIENDO EL QUE TENGA MAYOR CANTIDAD. NOS QUEDAMOS CON EL MÁXIMO.

NOTA

COMPLEJIDAD: COMO $m = n$

* CORRER DFS: $O(n+m) \stackrel{\swarrow}{=} O(n)$

* RECORRER CADA ARISTA DEL ARBOL PARA ETIQUETARLA, Y CONTAR CADA ARISTA BACK-EDGE $O(n)$

* RECORRER LOS CICLOS Y VER SU LONGITUD BUSCANDO EL MAXIMO $O(m) = O(n)$

LA COMPLEJIDAD RESULTANTE SERÁ:

$O(n) + O(n) + O(n) = O(n)$ COMO FUE PEDIDA.

NOTA

HOJA N° 4

FECHA

4) RECORRO TODAS LAS ARISTAS E IDENTIFICO LAS DE PESO NEGATIVO. LUEGO LAS ELIMINO DEL GRAFO Y CORRO DIJKSTRA DESDE s Y DESDE t (INVIRTIENDO LAS ARISTAS). ASÍ OBTENGO $d_{min}(s,w)$ Y $d_{min}(w,t)$ $\forall w \in V(G)$.

TENIENDO EL COSTO DEL CAMINO MÍNIMO DE s a t EN LA PRIMERA VEZ QUE CORRÍ DIJKSTRA, PUEDO IR ANALIZANDO SI PUEDO AGREGAR UNA ARISTA A ESE CAMINO MÍNIMO, SI TIENE MENOR COSTO (SI ENCUENTRO UNO MEJOR)

PARA ESO, TENIENDO EL CONJUNTO DE ARISTAS NEGATIVAS QUE SAQUÉ, LLAMÉMOSLO N, QUIERO QUEDARME CON:

$$\otimes \; \min \left\{ d(s,t),\ d(s,u) + c(u \to v) + d(v,t) \right\} \quad \forall\ (u \to v) \in N$$

MÍNIMO ENTRE EL CAMINO MÍNIMO DE s a t SIN ARISTAS NEGATIVAS Y EL COSTO DE LOS CAMINOS POSIBLES AGREGANDO ALGUNA DE LAS ARISTAS NEGATIVAS.

VEAMOS SU COMPLEJIDAD:

- BUSCO ARISTAS NEGATIVAS Y LAS SACO DEL GRAFO EN $O(m)$
- CORRO DIJKSTRA DESDE s Y DESDE t EN $O(\min\{n^2, m \log m\})$
- RECORRO N CONJUNTO DE ARISTAS NEGATIVAS Y POR CADA UNA ANALIZO EL MÍNIMO CAMINO CON CHEQUEANDO $\otimes$ EN $O(m)$

LA COMPLEJIDAD RESULTANTE SERÁ

$$O(m) + O(\min\{n^2, m\log m\}) + O(m)$$
$$= O(\min\{n^2, m \log m\})$$

NOTA

HOJA N° 5

FECHA

AGREGO PARA INDICAR QUE TERMINÓ EL JUEGO

1) a) $G: \{1, n, n+1\}: \mathbb{N} \to \mathbb{N}$

$G(i,v)$ = MÍNIMA CANTIDAD DE TIEMPO QUE LE TOMA A PANCHE RESOLVER LOS NIVELES $i, \dots, n$ SI EMPIEZA CON v VIDAS

$$G(i,v) \begin{cases} 0 & \text{si } i = n+1 \\ G(i, 100n+1) & \text{si } v > 100n+1 \\ \min\{t_{ip} + G(i+1, v-p+v_i) \mid 0 \le p \le \min\{100, v-1\}\} & \text{c.c.} \end{cases}$$

- SI $i = n+1$ TERMINÓ EL JUEGO ASÍ QUE EL TIEMPO SERÁ 0.

- EN EL ENUNCIADO DICE QUE PANCHE PUEDE GASTAR A LO SUMO 100 VIDAS EN CADA NIVEL, ESTO NOS DICE QUE TENER $100n+1$ VIDAS ES LO MISMO QUE TENER ∞ VIDAS, YA QUE NUNCA VA A PODER GASTARLAS.

- QUIERE EL MÍNIMO TIEMPO POSIBLE DE RESOLVER EL NIVEL EN EL QUE ESTÁ GASTANDO p VIDAS, CON p TODOS LOS VALORES POSIBLES HASTA 100 QUE ES LA CANTIDAD MÁXIMA QUE PUEDO GASTAR, Ó HASTA $v-1$ QUE ES LA CANTIDAD DE VIDAS QUE TENÍA ANTES -1 PORQUE PUEDE TENER 1 VIDA POR LO MENOS PARA NO PERDER EL JUEGO, Y VER EL TIEMPO DEL SIGUIENTE NIVEL SI GANÉ V_i VIDAS EN ESTE Y PERDÍ p.

EL LLAMADO INICIAL DEL ALGORITMO ES $g(1, V_0)$ CON V_0 UNA CANTIDAD DE VIDAS POTENCIALMENTE MAYOR QUE 1.

b) VEAMOS SUPERPOSICIÓN DE SUBPROBLEMAS

CANTIDAD DE LLAMADOS RECURSIVOS: $\Omega(101^n)$ PORQUE A LO SUMO HAY 100 LLAMADOS RECURSIVOS POR NIVEL

CANTIDAD DE SUBINSTANCIAS:

$v = 1 \dots 101n+1$ → YA QUE EN C.C. v PUEDE SER A LO SUMO $100n+1$ Y LE ESTOY SUMANDO V_i QUE ES A LO SUMO n

$i = 1 \dots n+1$

$$O((101n+1) \times (n+1))$$
$$= O(n^2)$$

NOTA

ES FÁCIL VER QUE HAY MUCHOS MÁS LLAMADOS RECURSIVOS QUE CANTIDAD DE INSTANCIAS DIFERENTES, POR LO TANTO G TIENE LA PROPIEDAD DE SUPERPOSICIÓN DE SUBPROBLEMAS.

d) CONSIDEREMOS MATRIZ DE MEMOIZATION M DE TAMAÑO $(n+1) \times (100n+1)$ INICIALIZADA CON UN VALOR INDEFINIDO $\perp$

```
G(i,v):
    Si i = n+1 retornar 0
    Si v > 100n+1 retornar G(i, 100n+1)
    Si M[i][v] == ⊥:
        M[i][v] = ∞
        FOR (p = 0 ; p ≤ min(100, v-1); p++):
            M[i][v] = min(M[i][v], T[i][p] +
                                   G(i+1, v-p+V[i]))
    retornar M[i][v]
```

LA COMPLEJIDAD DE ESTE ALGORITMO ES

\# ESTADOS × TIEMPO PARA RESOLVERLOS

$$O(n^2) \times O(1) = O(n^2)$$

NOTA

Comentarios de retroalimentación	Ejercicio 1: A Ejercicio 2: A Ejercicio 3: A Ejercicio 4: A Nota final: Aprobado. Corrector: Francisco No tengo mayores comentarios; todo estaba muy claro.